Visual and tactile fusion for estimating the pose of a grasped object

David Álvarez[1]*[0000-0003-2056-640X], Máximo A. Roa[2][0000-0003-1708-4223], Luis Moreno[1][0000-0002-4450-680X]

[1] Systems Engineering and Automation Department, Carlos III University of Madrid, Spain, dasanche,moreno@ing.uc3m.es

[2] Institute of Robotics and Mechatronics, DLR - German Aerospace Center, Germany, maximo.roa@dlr.de

Abstract. This paper considers the problem of fusing vision and touch senses together to estimate the 6D pose of an object while it is grasped. Assuming that a textured 3D model of the object is available, first, Scale-Invariant Feature Transform (SIFT) keypoints of the object are extracted, and a Random sample consensus (RANSAC) method is used to match these features with the textured model. Then, optical flow is used to visually track the object while a grasp is performed. After the hand contacts the object, a tactile-based pose estimation is performed using a Particle Filter. During grasp stabilization and hand movement, the pose of the object is continuously tracked by fusing the visual and tactile estimations with an extended Kalman filter. The main contribution of this work is the continuous use of both sensing modalities to reduce the uncertainty of tactile sensing in those degrees of freedom in which there is no information available, as presented through the experimental validation.

Keywords: pose estimation, sensor fusion, tactile sensors, visual information

1 Introduction

In-hand object pose estimation is a natural cognitive online process that humans perform while grasping or manipulating objects. There are several indications that humans use complementary sensor information from vision and touch in this process [1, 2], that manipulation tasks rely on accurate and fast pose estimation [3], and that human memory is multi-sensorial in nature [4].

In computer vision, several methods for stable and reliable object pose tracking exist in literature. Many approaches are based on tracking object boundaries [5] or on non-linear pose computation using RGB-D information [6]. Texture tracking [7] and model-free cues [8] have also been presented. While simple scenarios allow an accurate object pose tracking, more complex tasks may require inferring object properties [9]. In this paper, a combination of texture tracking and image-based motion cues is used for processing visual information, inspired by [10].

Tactile sensing has also received a great deal of attention recently, including applications in texture and object recognition [11, 12] and in-hand pose estimation. Object recognition by exploring the object's surface and edges using a particle filter combined

with an Iterative Closest Point approach was presented in [13]. In the case of pose estimation, some methods use an offline description of the object's facets to match the current sensor measurements [14]. Also, preventing physically unfeasible solutions can be considered for the in-hand pose estimation process [15]. Position and torque measurements from the finger joints have been used to estimate the pose of the object as well as the contact state of the grasp [16]. Our approach for using tactile information, initially presented in [17], uses a particle filter to enhance solutions that match sensor measurements, thus avoiding physically unfeasible estimations.

One of the first attempts to integrate vision and touch was presented in [18], using geometric models of objects that are complemented with tactile sensing for gathering information on the unseen parts. More recently, RGB-D and tactile data were treated using an invariant extended Kalman filter (EKF) to discover and refine 3D models of unseen objects [19], with practical applications for simplified models of symmetric objects characterized by two features, width and angle. The fusion of tactile and visual measurements enables also the pose estimation of a moving target at high rate and accuracy [20]. Instead of tracking an external object, they follow a probe, which produces tactile measurements, mounted on an hydraulic manipulator. Fusion of tactile and visual information has been used to refine an initial estimation of the hand-object pose for grasping applications [21, 22]. Several approaches have tried to simultaneously use vision and tactile information for in-hand object pose estimation. However, the use of visual information often ends when the hand is closed around the object, and afterward only tactile information is used for the pose estimation process [23, 24]. In [25], tactile sensing is used to add physical constraints to a vision-based estimator; however, the pose estimation is mainly based on vision, therefore heavy occlusions are difficult to manage.

Both vision and touch can be used to separately estimate the 6-DOF pose of an object, but typically each estimation is not accurate along one or several degrees of freedom. This work is centered on the effective combination of both modalities to improve the pose estimation during a grasping action. The visual estimation is based on [10], which uses a CAD-based pose estimation and an optical flow-based tracker, while the tactile information is processed following our previous work in [17]. The fusion of both estimations is done using an Extended Kalman Filter, which prioritizes one of the sensor modalities depending on the accuracy of each method at a given stage. The visual information is constantly used to complement the information gathered by tactile sensors while there is contact with the object, thus reducing uncertainty along the directions where the tactile information does not provide enough information to effectively estimate the pose of the object.

2 Sensor Fusion Framework

The grasp execution is divided into different phases depending on the existence or not of contact between hand and object. These phases define the type of information available for estimating the hand-object pose, as summarized in Table 1. During the pre-grasp phase, the hand moves towards the object to achieve the pose from which the grasp is executed. The vision system has a clear view of the scene, while there is no useful tactile

information yet. The grasp phase starts when the first contact between hand and object is detected by the force sensors of the hand, and ends when the hand is commanded to open the fingers. It is during this phase that the two sensing modalities can be independently used for estimating the hand-object pose; however, the vision system may have difficulties tracking the object due to occlusions created by the fingers wrapped around the object. Finally, when the hand releases the object there is only visual information available, although there might be no estimation at all when the vision system gets lost. Therefore, in the case of the first and third phases of the grasp execution, only visual information is used in the pose estimation, while in the second one, both visual and tactile information are effectively fused.

Table 1. Information provided by the sensors in each grasp phase.

Grasp Phase	Pre-grasp (Before closing)	Grasp (Object in hand)	Release (Hand open)
Information	Visual	Visual/Tactile	Visual

2.1 Vision-based estimation

The vision-based pose estimation and visual tracking are both based on Simtrack [10]. In our case, the tracking problem is simplified since we only use RGB information provided by the vision system. The system assumes that a 3D model of the object, which includes texture, is available and used to perform a comparison with the information provided by the camera.

From the color feed provided by the RGB camera, SIFT key-points (Scale-Invariant Feature Transform) [26] and optical flow (movement in the image) are extracted. The SIFT keypoints are used by the object detection system in order to provide an initial estimation for starting the tracker start, or for restarting it when the tracker is not capable anymore of following the object. On the other hand, motion cues provided by optical flow are used by the tracking system. Note that the original software also uses cues extracted from stereo disparity or depth information, but in this case, this information is not available due to the experimental setup (Section 3). The tracker updates the estimated pose of the object so that the consistency between the motion and a 3D representation of the environment is maximized.

When the pose estimation is active, GPU libraries are used in order to extract the SIFT features from the 2D images. Then, a Random Sample Consensus (RANSAC) method [27], which tries to match SIFT descriptors extracted from the RGB images to the textured 3D models in the database, is used to find correspondences and extract the 6-DOF pose of the object. This step tries to perform an exhaustive matching in the given frame and, therefore, this sparse estimation does not depend on the previous one, or on the movement in the images.

Once an initial estimation is available, the tracker starts and uses motion cues to compute the motion in the scene (using GPU libraries [28]). For this, an Augmented Reality (AR) version of the estimation is rendered. Then, optical flow is computed out of the difference between the (partially) synthetic image, rendering the object model based

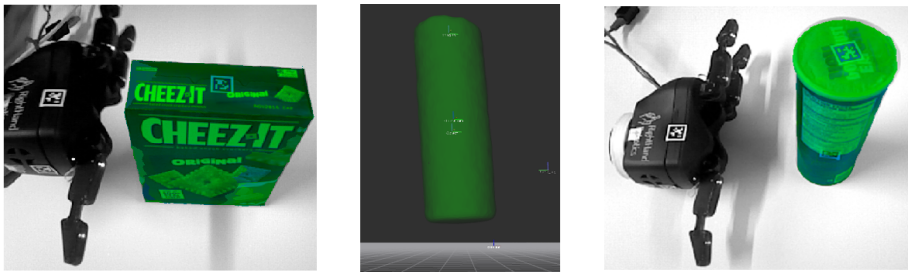

Fig. 1. Examples of the initial visual estimation for the object's pose. A common error that appears is that the estimated object pose is floating above the supporting table, as illustrated in the central image.

on the current pose estimate, and the next obtained image. When used for tracking, this information is insensitive to drift since it measures the difference between the current scene hypothesis and the observed scene (rather than simply the image motion). For the same reason, it can be used to measure the reliability of tracking. The motion observed by the optical flow is used to recover a rigid rotation and translation that best explains the visual cues, and transforms the pose estimation accordingly.

Fig. 1 shows examples of the initial estimation based on visual information. The estimated pose is also shown (in green) on the image. The image in the center shows an error that appears commonly, namely, the estimated pose is too high over the table surface, which is not physically realistic. This is due to the point of view used for acquiring the images. Average errors of the initial pose estimated by the vision system for two different objects are presented in Table 2.

Table 2. Average error in the initial visual estimation.

Object	Can			Box	
Trial	1	2	3	1	2
Error in position (cm)	3.67	2.67	1.40	3.08	2.97
Error in orientation (°)	0.45	1.52	1.03	0.97	0.85

2.2 Tactile-based estimation

Tactile-based estimation is only possible when there is at least one contact between the hand and the object. Therefore, when the first contact is detected, a particle filter starts looking for object poses using, as first prior estimation, the last pose provided by the vision system. The estimated pose should agree with the information provided by the tactile sensors.

The reference frame used for the tactile pose estimation is located at the wrist of the hand. The parameters describing the object pose are

$$x = [q,t]^T = [q_x, q_y, q_z, q_w, t_x, t_y, t_z]^T \tag{1}$$

where q is a rotation expressed as a quaternion, and t is a translation vector.

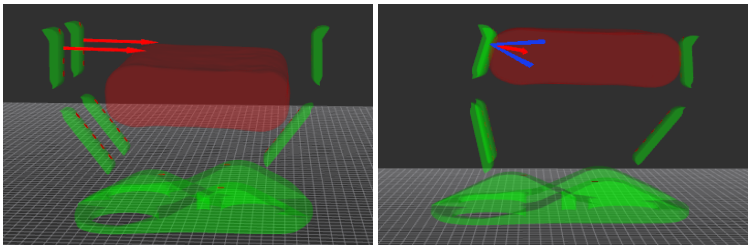

Fig. 2. Left: two contacts are detected (red arrows), but the estimated pose of the object does not produce contacts. Right: friction cone at a contact location.

Assuming that the 3D model of the object is available, the estimation is tackled by combining the following general ideas:

1. When a contact is detected by a sensor, the estimated object pose must produce a contact at the same location. Fig. 2 (left) illustrates a case where the estimated object pose cannot explain the contact readings in two fingers.
2. The estimated object pose should not be in collision with the hand (just in contact). Besides, the object cannot float without contacting the hand at all when at least one sensor reading is positive.
3. The inward normal of the object surface at the contact location and the outward normal at the contact surface in the hand should have the same direction. When friction is considered, the normals do not necessarily have to be aligned, but, since the friction coefficient is in general not known, the angle between normals should be as close to 0 as possible (right side of Fig. 2).

A deeper explanation on the implementation of the particle filter (where each particle represents one pose of the object) can be found in [17]. The general outline of the algorithm is shown in Algorithm 1. The key ideas presented above are included in the measurement model that weighs the estimation of a given particle as explained below.

Algorithm 1 Bootstrap Particle Filter

procedure BPF($N_p, prior\_estimation$)
Initialization:
  $x_i(0) \sim Pr(x(0)),\ \ W_i(0) = \frac{1}{N_p}$
Importance Sampling:
  $x_i(t) \leftarrow system\_model(x_i(t-1), input_t)$
  $W_i \sim Pr(W_i(t))$
Weight Update:
  $W_i(t) = W_i(t-1) \times measurement(y(t)|x_i(t))$
Weight Normalization:
  $W_i(t) = \frac{W_i(t)}{\sum_{i=1}^{N_p} W_i(t)}$
Resampling:
  $if\ \hat{N}_{eff}(t) \leq N_{thresh},\ then\ \hat{x}_i(t) \Rightarrow x_j(t)$
end procedure

Weight Update: measurement model The measurement model gives to each one of the particles a weight that quantifies how similar is the state expressed by that particle to the true state of the object, comparing the estimation with the measurements provided by the position and tactile sensors in the hand. The measurement model used in this work is inspired by [15]. Three new features have been added: first, not only the sensor location but also the force measurements are used to compute the real contact locations. Second, the evaluation method considers differently each sensor depending on whether it is in contact or not. And third, the friction cone of a contact is considered to evaluate the feasibility of a contact between the hand and the object.

In order to evaluate the particles, the scene is simulated using the Flexible Collision Library (FCL) [29] to compute the shortest distance (no collision, positive value) or deepest penetration (in collision, negative value) between each sensor and the object. Taking into account this information, three kind of measurements are considered:

- For each sensor that does not detect contact with the object, a probability is assigned to each particle based on its distance to the object d_i^o by:

$$p_{nc,i}(d_i^o) = 0.5 * \left(1 + erf\left(\frac{d_i^o}{\sqrt{2}\sigma_{nc}}\right)\right) \tag{2}$$

  where σ_{nc} is a standard deviation value that can be adjusted to match the inaccuracy of the measurements, and erf corresponds to the error function. This function is chosen to assign high weights to positive distances and small weights to negative distances, which helps to avoid estimations that predict unreal collisions.
- For each sensor that detects a contact, the distance d_i^o is used to associate a probability to the measurement with:

$$p_{c,i}(d_i^o) = e^{-0.5\left(\frac{d_i^o}{\sigma_c}\right)^2} \tag{3}$$

  where σ_c can be adjusted to account for the uncertainty in the force sensors. This function assigns high weights to values that are close to zero, i.e. close to contact.
- Assuming the grasp is stable, the normal of the surface of the object at the contact point (for the sensors that detect a contact) should lie within the friction cone around each contact point in the hand. The contact force measured by the hand is considered to be normal to its surface, therefore, the angle α_i between the normals to the surfaces can be computed, and afterward evaluated with:

$$p_{a,i}(\alpha) = e^{-0.5\left(\frac{\alpha_i}{\sigma_a}\right)^2} \tag{4}$$

  where σ_a accounts for the friction between the surfaces. This function assigns high weights to values that are close to 0.

Finally, a combined weight for each particle (W_i) can be expressed as:

$$W_i = \prod_{k=1}^{N_m} p_{nc,i} * p_{c,i} * p_{a,i} \tag{5}$$

where N_m is the number of measurements for each particle. This weight is calculated for every particle during the update step in Algorithm 1.

2.3 Sensor Fusion with Extended Kalman Filter

The Extended Kalman filter is a linearized version of the Kalman Filter, a recursive continuous state observer that uses knowledge of the system and measurement models and their corresponding noises. These models can be formulated as:

$$\begin{aligned} X_t &= f_s(X_{t-1}, U_t, V_t) \\ Z_t &= f_m(X_t, W_t) \end{aligned} \tag{6}$$

where f_s is the function that defines the system dynamics, computing the current state X_t based on the value of the previous state X_{t-1} and the input U_t, and V_t represents the noise of this function. Furthermore, f_m is the function that defines the measurement system, computing the current sensor readings Z_t based on the value of the actual state X_t, and W_t represents the noise of this function. Both V_t and W_t are considered to be discrete functions representing a zero mean Gaussian disturbance, with Q and R as their respective covariances.

Using this knowledge and eq. (1) as the state of our system, the Extended Kalman filtering process is divided into two steps:

- Prediction step: uses a previously estimated state ($\hat{X}_{t-1}$), the input (U_t) and the system model (f_s) to predict the value of the next state, as well as the state-estimated covariance:

$$\begin{aligned} U_t &= [\Delta q_t, \Delta t_t]^T \\ \hat{X}_{t|t-1} &= f_s(X_{t-1}, U_t, V_t) = [q_{t-1} * \Delta q_t * q_v, t_{t-1} + \Delta t_t + t_v] \\ P_{t|t-1} &= \left(\frac{\partial f_s}{\partial x}\right) P_{t-1|t-1} \left(\frac{\partial f_s}{\partial x}\right)^T + Q \end{aligned} \tag{7}$$

  where $\left(\frac{\partial f_s}{\partial x}\right)$ is the Jacobian of f_s with respect to state X, and $P_{t|t-1}$ is the estimated covariance. U_t is measured as the average movement of the fingers that are in contact with the object.
- Update step: uses the current sensor measurements (visual and tactile estimations) together with the statistical properties of the model to correct the initial estimate. Besides, the Kalman gain and state-estimate covariance are also computed.

$$\begin{aligned} K_t &= P_{t|t-1} \left(\frac{\partial f_m}{\partial x}\right)^T \left[\left(\frac{\partial f_m}{\partial x}\right) P_{t|t-1} \left(\frac{\partial f_m}{\partial x}\right)^T + R\right]^{-1} \\ \hat{X}_{t|t} &= X_{t|t-1} + K_t(Z_t - f_m(\hat{X}_{t|t-1})) \\ P_{t|t} &= \left[I - K_t \left(\frac{\partial f_m}{\partial x}\right)\right] P_{t|t-1} \end{aligned} \tag{8}$$

  where K_t is the Kalman gain, $\left(\frac{\partial f_m}{\partial x}\right)$ is the Jacobian of f_m with respect to state X, and $P_{t|t}$ is the covariance of the estimation. Since the measured properties (Z_t) are of the same type as the predicted estate of the system (X_t), position and orientation, f_m is just the vertical concatenation of two 6x6 identity matrices.

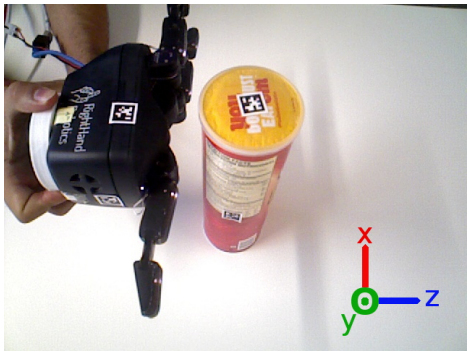

Fig. 3. Experimental setup for object pose estimation. A reference frame parallel to the wrist reference frame is shown in the lower right corner.

These two steps are repeated for every sample: $t = 1, 2, \ldots, T$.

The only user-configured parameters of the algorithm are the covariance matrices representing the system and sensor noise. Since we are fusing information coming from two different sensors, it is important to carefully choose the sensor noises, since the Extended Kalman filter naturally gives more importance to the signal measured by the sensor with less noise, i.e., the more reliable one. In order to choose these values, the tactile and visual readings were studied offline separately, computing average and standard deviation errors of their estimation. Noise values have been chosen to be $1/100$ of the average noise in each axes, giving as a result that noise in the tactile estimation is 2 times larger for the orientation values, 1.3 larger in the Y axis, and 2.5 smaller in the X and Z axes.

Finally, the filter is executed every time there is a new reading from any of the sensors (estimations); since it is possible that not all of them are available at the same time, the last available reading is always used. This is also applied if any of the estimators (visual or tactile) loses track of the object.

3 Setup Description

For the experimental tests, we use the ReFlex TakkTile hand (Fig. 3). The hand is equipped with two types of sensors: pressure sensors located along the fingers (9 per finger) and the palm (11 sensors), and magnetic encoders in the proximal and distal joints, which allow computing the location of both phalanges in each finger. The position of each tactile sensor and the normal vector to the surface at its position can be constantly computed. The provided force measurements are based on the pressure transmitted by the rubber that covers the fingers. However, since the pressure flows through the rubber, one single contact with an object may be detected by two (or more) consecutive sensors. When this happens, a linear combination of measurements is performed to compute the actual contact location c_i with respect to the wrist, as follows:

$$c_i = t_i + \left| 1 - \frac{f_i}{f_i + f_{i+1}} \right| \times (t_{i+1} - t_i) \tag{9}$$

Fig. 4. Cheez-it box grasped from the side with the robotic hand held by a human operator. The sequence is ordered from left to right, and includes the pre-grasp pose, grasping, moving and releasing the object.

where t_i is the position of sensor i and f_i its corresponding force measurement. Note that this model assumes that there is maximum one contact with the object at each link of the fingers.

The camera used to retrieve visual information is an RGB camera with a resolution of 640x480 at 30 frames per second. The spatial location of the hand with respect to the camera is provided by Apriltags [30] located on the top surface of the hand, as shown in Fig. 3. Note that the object also has an Apriltag on the top surface, which is used to compute the ground truth for the relative pose of the object with respect to the hand.

For the measurements, a human operator holds the hand and approaches the object, grasping it and releasing it. The hand could also be attached to a robotic arm; this has no influence on the results of the estimation of the hand-object relative pose nor on the in-hand object pose estimation. The performed experiments follow these steps:

- Hand and object are placed on a flat surface, and the scene is perceived with the RGB camera. The visual estimator starts looking for the object.
- An operator picks up the hand and positions it to execute the grasp.
- After 10 seconds, the fingers close toward the object. Once they are in contact, a constant closing velocity is maintained in all fingers to make the grasp stable.
- The tactile estimation is started as soon as the first contact between hand and object is detected. An initial population of particles is built (adding Gaussian noises) based on the last estimation given by the vision system. It is ended when the hand is commanded to open.
- The object is lifted by the operator. As long as both tactile and visual estimations are available, the extended Kalman filter computes the in-hand object pose.

4 Experimental Results

Two different objects from the YCB database [31] are used for the tests, a Pringles can and a Cheez-it box (Fig. 1). For the initial test of the concept, three test sequences were made with the can and two with the box. For each of them, pose estimation tests were run 5 times. One of the test runs can be seen in Fig. 4. From left to right, the figure shows the pre-gras pose, grasping pose, lifting and moving the object, and hand opening.

Fig. 5 shows the results of the pose estimation for one of the experiments using the Cheez-it box. For the displacements in the three axes, the ground truth (GT - light blue), the tactile (red), visual (dark blue) and the EKF-based fusion estimation (green) are shown. The orientation error around the three axis is also shown in the figure. Note

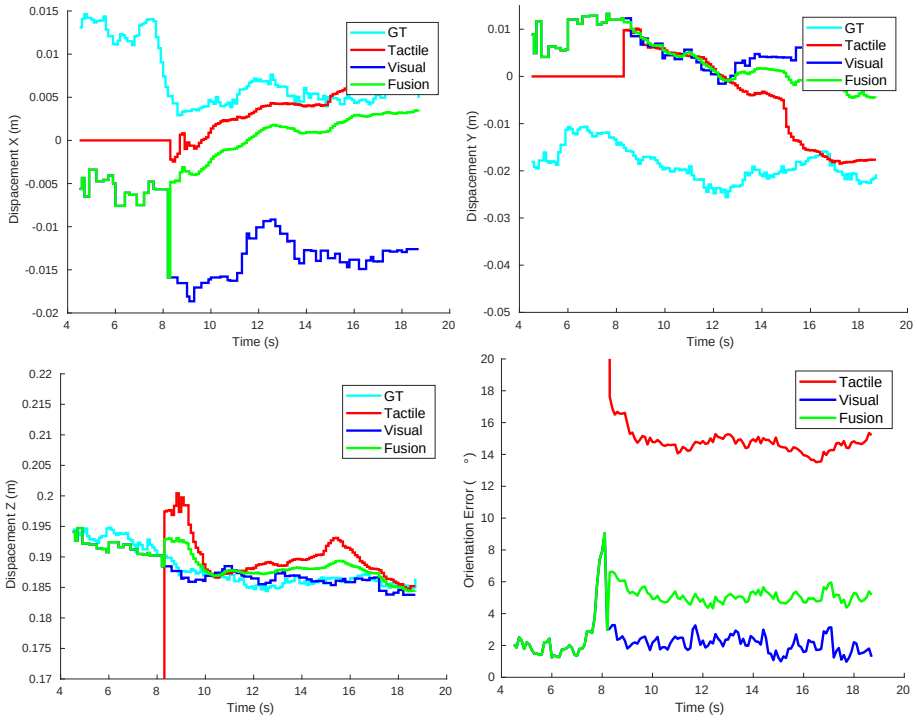

Fig. 5. Evolution of the displacement and orientation error while grasping the Cheez-it box with the sequence shown in Fig. 4.

that at the beginning of the movement, the EKF-based estimation is the same as the vision-based one, since there is no contact with the object yet. The first contact between the hand and the object is detected at about 8s. From that moment, the estimation using the fusion technique here presented differs from the pose estimations using only one sensing modality. In the X axis (top-left of Fig. 5), the estimation is corrected by the influence of the tactile system, while in the Z axis, the estimation is better for the vision system (tactile information does not help to pinpoint the object location along this axis for this particular object). It is possible to appreciate the initial error in the estimation of the location along the Y axis due to the vision system, which remains almost constant with time. Because the tactile system has no means of measuring changes in the Y axis and has less accuracy in estimating the changes in orientation, the resulting estimation in that case follows more closely the estimation coming from the vision system (right side of Fig. 5). Note that this is a result of the matrices used in this technique to represent the noise for each sensor modality.

In the case of the orientation error suffered by the tactile system, it is related to the initial error found along the Z axis. Because a contact between the box and the palm of the hand is detected, but the initial estimation in the Z axis is actually off by almost 2cm, the simulation recreates the same situation by turning the box around the X axis so that this same contact is detected. However, these errors are successfully corrected over time by the fusion of the two estimations.

Table 3. Average errors and standard deviations in the pose estimation for the selected objects in different test sequences.

Object	Trial	Vision		Tactile		Fusion	
		μ	σ	μ	σ	μ	σ
Can	1	3.72cm	0.9	2.04cm	0.5	3.15cm	0.8
		0.13°	0.8	2.32°	0.6	1.00°	0.8
	2	4.03cm	1.2	3.47cm	0.61	3.55cm	1.1
		7.82°	6.4	13.90°	3.3	9.65°	6.3
	3	3.03cm	0.3	3.58cm	0.4	3.18cm	0.3
		2.71°	0.6	6.54°	2.3	5.28°	1.5
Box	1	2.29cm	0.2	1.69cm	0.7	1.78cm	0.4
		0.64°	0.8	11.55°	2.3	2.2°	1.6
	2	3.01cm	0.7	1.92cm	0.6	2.03cm	0.6
		4.15°	2.1	7.8°	2.5	4.20°	2.2

Table 3 shows the average and standard deviations for the errors obtained in the different test sequences. The first test of the box corresponds to the one shown in Fig. 5. The errors are computed as an average over all the grasping action for the fused estimation, including a short time before grasping has occurred and after the object has been released. The same period is used for computing the errors for the vision system. However, for the tactile system the error only covers those moments in which there is effective contact between object and hand.

In the second test of the box and the can, the visual estimation is misled by the movement of the fingers, and this is estimated as a movement of the object itself. This results in larger errors both in position and orientation. In the third experiment of the can, the vision system is not able to track the object while it is grasped by the hand because of the occlusion of the object, and it is not able to recover until the fingers open again. This results in a worse estimation in general, first because the prior given to the tactile system is worse, but also because there are no visual corrections in those axes where the tactile system is weaker. Lastly, the first experiment conducted with the can produces very good estimations, the only error found is given by the initial error committed by the visual estimation. The magnitude of the errors described in Table 3 are in the same range of those found in similar works [23], [19], [25], and there is a clear improvement in the initial visual estimation used in our work.

5 Conclusions

This paper presented a 6D object pose estimation method that combines visual and tactile information. The fusion of the information provided by both sensing modalities is performed by an extended Kalman filter. An initial experimental evaluation with real data captured with an RGB camera and a robotic hand is performed to study the integration of the two complementary sensor modalities in order to successfully reduce the overall uncertainty of the pose estimation.

Improvements to the approach presented here include a better initial visual estimation, since this error is later propagated to the fusion with the tactile information. A

tracking system more robust to object occlusions would also be desirable, and experiments with objects of more complex geometries is a next step. A possible extension of this work could investigate how to avoid using explicit object models in the estimators.

Acknowledgments

The authors want to thank Naiara Escudero for her assistance on the implementation of the Extended Kalman Filter, and Karl Pauwels for insights given on the use of Simtrack.

The research leading to these results has received funding from RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub, S2018/NMT-4331, funded by Programas de Actividades I+D en la Comunidad de Madrid and co-funded by Structural Funds of the EU. This work has also received funding from the Spanish Ministry of Economy, Industry and Competitiveness under the project DPI2016-80077-R.

References

1. Lacey, S., Sathian, K. Visuo-haptic multisensory object recognition, categorization, and representation. Front Psychol., 5, 730 (2014).
2. Macura, Z., Cangelosi, A., Ellis, R., Bugmann, D., Fischer, M., Myachykov, A.: A cognitive robotic model of grasping. In: Int. Conf. Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, pp. 89–96 (2009).
3. Dogar, M., Hsiao, K., Ciocarlie, M., Srinivasa, S.: Physics-based grasp planning through clutter. In: Robotics: Science and Systems VIII (2012).
4. Vasconcelos, N., Pantoja, J., Belchior, H., Caixeta, F.V., Faber, J., Freire, M.A., Cota, V.R., de Macedo, E.A., Laplagne, D.A., Gomes, H.M., Ribeiro, S.: Cross-modal responses in the primary visual cortex encode complex objects and correlate with tactile discrimination. In: Proc. National Academy of Sciences, 108(37), pp. 15408–13 (2011).
5. Petit, A., Marchand, E., Kanani, K.: A robust model-based tracker combining geometrical and color edge information. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 3719–3724 (2013).
6. Choi, C., Christensen, H. I.: RGB-D object tracking: A particle filter approach on GPU. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 1084-1091 (2013).
7. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3D tracking using online and offline information. IEEE Trans. Pattern Analysis and Machine Intelligence, 26, 1385-1391 (2004).
8. Kyrki, V., Kragic, D.: Integration of model-based and model-free cues for visual object tracking in 3D. In: IEEE Int. Conf. Robotics and Automation, pp. 1566–1572 (2005).
9. Güler, P., Bekiroglu, Y., Pauwels, K., Kragic, D.: What's in the container? Classifying object contents from vision and touch. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 3961-3968 (2014).
10. Pauwels, K., Ivan, V., Ros, E., Vijayakumar, S.: Real-time object pose recognition and tracking with an imprecisely calibrated moving RGB-D camera. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 2733–2740 (2014).
11. Jamali, N., Sammut, C.: Majority voting: material classification by tactile sensing using surface texture. IEEE Trans. on Robotics, 27(3), 508–521 (2011).
12. Madry, M., Bo, L., Kragic, D., Fox, D.: ST-HMP: unsupervised spatio-temporal feature learning for tactile data. In: IEEE Int. Conf. Robotics and Automation, pp. 2262–2269 (2014).
13. Aggarwal, A., Kirchner, F.: Object recognition and localization: the role of tactile sensors. Sensors, 14, 3227–3266 (2014).

14. Haidacher, S., Hirzinger, G.: Estimating finger contact location and object pose from contact measurements in 3-D grasping. In: IEEE Int. Conf. Robotics and Automation, pp. 1805–1810 (2003).
15. Chalon, M., Reinecke, J., Pfanne, M.: Online in-hand object localization. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 2977–2984 (2013).
16. Pfanne, M., Chalon, M.: EKF-based in-hand object localization from joint position and torque measurements. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 2464-2470 (2017).
17. Álvarez, D., Roa, M.A., Moreno, L.: Tactile-Based In-Hand Object Pose Estimation. ROBOT 2017: Third Iberian Robotics Conference. Advances in Intelligent Systems and Computing, vol 694, pp. 716-728. Springer (2018).
18. Allen, P.K.: Integrating vision and touch for object recognition tasks. Int. J. of Robotic Research, 7, 15–33 (1988).
19. Ilonen, J., Bohg, J., Kyrki, V.: Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing. Int. J. Robotic Research, 33(2), 321–341 (2014).
20. Alkkiomäki, O., Kyrki, V., Kälviäinen, H., Liu, Y., Handroos, H.: Complementing visual tracking of moving targets by fusion of tactile sensing. Robotics and Autonomous Systems, 57, 1129–1139 (2009).
21. Kolycheva, E., Kyrki, V.: Task-specific grasping of similar objects by probabilistic fusion of vision and tactile measurements. In: IEEE-RAS Int. Conf. Humanoid Robots, pp. 704-710 (2015).
22. Zhang, M.M., Detry, R., Matthies, L., Daniilidis, K.: Tactile-Vision Integration for Task-Compatible Fine-Part Manipulation. In: Robotics: Science and Systems. Workshop on Revisiting Contact - Turning a Problem into a Solution (2017).
23. Bimbo, J., Rodríguez-Jiménez, S., Liu, H., Song, X., Burrus, N., Senerivatne, L. D., Abderrahim, M., Althoefer, K.: Object pose estimation and tracking by fusing visual and tactile information. In: IEEE Int. Conf. Multisensor Fusion and Integration for Intelligent Systems, pp. 65-70 (2012).
24. Bimbo, J., Seneviratne, L., Althoefer, K., Liu, H.: Combining touch and vision for the estimation of an object's pose during manipulation. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 4021–4026 (2013).
25. Schmidt, T., Hertkorn, K., Newcombe, R., Marton, Z., Suppa, M., Fox, D.: Depth-based tracking with physical constraints for robot manipulation. In: IEEE Int. Conf. Robotics and Automation, pp. 119-126 (2015).
26. Wu, C.: SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). http://github.com/pitzer/SiftGPU.
27. Lepetit, V., Fua, P.: Monocular model-based 3D tracking of rigid objects. Foundations and Trends in Computer Graphics and Vision, 1, (2005).
28. Pauwels, K., Tomasi, M., Diaz Alonso, J., Ros, E., Van Hulle, M.: A comparison of FPGA and GPU for real-time phase-based optical flow, stereo, and local image features. IEEE Trans. on Computers, 61(7), 999-1012 (2012).
29. Pan, J., Chitta, S., Manocha, D.: FCL: a general purpose library for collision proximity queries. In: IEEE Int. Conf. Robotics and Automation, pp. 3859–3866 (2012).
30. Olson, E.: AprilTag: a robust and flexible visual fiducial system. In: IEEE Int. Conf. Robotics and Automation, pp. 3400–3407 (2011).
31. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.: The YCB object and model set: towards common benchmarks for manipulation research. In: IEEE Int. Conf. Advanced Robotics, pp. 510–517 (2015).